TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

PASCHIMANCHAL CAMPUS
LAMACHAUR, POKHARA

SOFTWARE REQUIREMENT SPECIFICATIONS
on

ePurano

BY

SEAMOON PANDEY - PAS078BCT035
RAKSHA RAUT- PAS078BCT028

TO

Department of Computer and Electronics Engineering
POKHARA, NEPAL
June 8, 2024

Table of Contents

Chapter 1: Introduction

1.1 Background Theory

The ePurano application is designed as a platform to facilitate peer-to-peer transactions of secondhand items. Recognizing the growing demand for sustainable and cost-effective alternatives to purchasing new products, ePurano aims to create a seamless, user-friendly experience for both buyers and sellers in the secondhand market. The application will provide functionalities such as item listings, user profiles, secure messaging, payment processing, and review systems to ensure trust and transparency between users.
This Software Requirements Specification (SRS) document outlines the comprehensive requirements for the development of ePurano, detailing functional and non-functional requirements, system architecture, user interfaces, and the technological stack. The objective is to ensure a clear understanding of the application's capabilities, constraints, and performance expectations, providing a solid foundation for the development and deployment phases.
ePurano's mission is to empower individuals to efficiently buy and sell pre-owned goods, fostering a community-driven marketplace that promotes sustainability and economic efficiency. This document serves as a roadmap to guide the project team in delivering a robust and reliable platform that meets the needs of its users while adhering to industry standards and best practices.

1.2 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of ePurano, a platform designed to facilitate peer-to-peer transactions of secondhand items. This document aims to ensure a clear understanding of the application's capabilities, constraints, and performance expectations, providing a solid foundation for the development and deployment phases.

1.3 Scope

ePurano is an application platform that enables users to buy and sell secondhand items in a seamless, user-friendly environment. The application will provide functionalities such as item listings, user profiles, secure messaging, payment processing, and review systems to ensure trust and transparency between users. The goal is to empower individuals to efficiently buy and sell pre-owned goods, fostering a community-driven marketplace that promotes sustainability and economic efficiency.

1.4 Overview

The ePurano platform will be developed using the Waterfall software development model, a linear and sequential approach ideal for projects with well-defined requirements. This model ensures systematic execution through the following phases:

- **Requirements Analysis** Gather and document all requirements for the ePurano platform.
- **System Design** Outline the architecture and detailed design based on the requirements.
- **Implementation** Develop the platform according to the design specifications.
- **Integration and Testing** Conduct thorough testing to ensure functionality and reliability.

Waterfall model follows a sequential approach, with each phase of the SDLC completed before moving to the next phase.

Chapter 2: Overall Description

2.1 Product Perspective

ePurano is a new application designed to create a marketplace for peer-to-peer transactions of secondhand items. Unlike traditional e-commerce platforms, ePurano focuses specifically on pre-owned goods, promoting sustainability and cost savings for users. The platform will be standalone but will integrate with various third-party services for payment processing, user authentication, and messaging.

2.2 User Functions

- **User Registration and Authentication**: Users can create accounts, log in, and manage their profiles.
- **Item Listings**: Users can list items for sale, including uploading images and descriptions.
- **Search and Filter**: Buyers can search and filter listings by categories, price range, and location.
- **Secure Messaging**: Users can communicate securely within the platform to negotiate and finalize transactions.
- **Payment Processing**: Integrated payment gateway for secure transactions.
- **Review System**: Users can leave reviews and ratings for each other to build trust within the community.

2.3 Admin Functions

- **User Management**: Admins can manage user accounts, including registration approvals, suspensions, and deletions.
- **Content Moderation**: Admins can review and moderate listings, messages, and reviews to ensure compliance with platform policies.
- **Platform Analytics**: Admins can access and analyze platform usage data to improve services.
- **Customer Support**: Admins provide support to users, addressing any issues or concerns.

2.4 User Characteristics

2.4.1 Users Characteristics

- **Sellers**: Individuals looking to sell secondhand items in a user-friendly and reliable environment.

- **Buyers**: Individuals seeking to purchase secondhand goods at competitive prices.

2.4.2 Administrators

- **Platform Managers**: Staff responsible for managing the platform, handling user support, and ensuring smooth operation.

2.5 Constraints

Key constraints for the development and operation of ePurano include:

- **Budget**: Limited funding for development and initial marketing.
- **Time**: Tight deadlines to launch the MVP (Minimum Viable Product).
- **Technological**: Integration with third-party services must comply with their APIs and terms of service.
- **Regulatory**: Compliance with data protection regulations and e-commerce laws.

2.6 Assumptions and Dependencies

- **User Adoption**: Successful adoption by users is critical for the platform's success.
- **Third-Party Services**: Dependence on external services for payment processing, user authentication, and messaging.
- **Internet Access**: Users must have internet access to use the platform.
- **Device Compatibility**: The platform must be compatible with modern web browsers and mobile devices.

By considering these aspects, the development team can ensure that ePurano meets the needs of its users and operates smoothly within its constraints and dependencies.

Chapter 3: Specific Requirements

3.1 Interface Requirements

Interface requirements define how the system interacts with users and other systems.For ePurano, these would include:

3.1.1 User Interfaces

- **Homepage** A clean, attractive homepage with links to major sections such as categories, search bar, user login/register, and featured listings.
- **Listing Page** Detailed view of an item, including photos, description, price, seller information, and contact options.
- **User Dashboard** Access to user's listings, messages, profile settings, and transaction history.
- **Search Interface** A user-friendly search bar with filters and sorting options.

3.1.2 Admin Interfaces

- **Moderation Tools** Interface for administrators to manage user reports, edit or remove inappropriate listings, and handle user bans.
- **Analytics Dashboard** Overview of platform metrics such as user activity, listings, and transaction volumes.

3.1.3 External Interfaces

- **Payment Gateway** Secure integration with third-party payment processors.
- **Email service** Integration with an email service provider for sending notifications, password resets, and other communications.
- **Social Media Integration** Options for users to log in using their social media accounts and share listings on social media platforms.

3.2 Functional Requirements

Interface requirements define how the system interacts with users and other systems.

3.2.1 User Registration and Authentication

- Users must be able to create accounts using email or social media logins.
- must be able to log in and log out securely.
- Users should be able to reset their passwords if forgotten.

3.2.2 User Profile

- Users should have profiles where they can manage their personal information and preferences.
- Users should be able to view their own buying and selling history.

3.2.3 Listing items for Sale

- Users must be able to create new listings, including adding photos, descriptions, categories, and pricing.
- Users should be able to edit or delete their listings.
- Users should be able to edit or delete their listings.
- Users should be able to mark items as sold.

3.2.4 Browsing and Searching

- Users must be able to browse items by categories.
- Users should be able to use a search function to find specific items using keywords, filters (price range, category, condition), and sort options (newest, price, etc.).

3.2.5 Payment Integration

- The platform should support secure payment methods for online transactions.
- Users should be able to track the status of their payments.

3.3 Non-Functional Requirements

Non-functional requirements define the system's operation rather than specific behaviors.

3.3.1 Performance

- The website should load within 3 seconds on a standard broadband connection.
- The system should handle at least 10,000 concurrent users without performance degradation.

3.3.2 Scalability

- The platform should be able to scale to accommodate growing numbers of users and listings.

3.3.3 Reliability

- Data backups should be performed daily to prevent data loss.

3.3.4 Security

- ser data should be encrypted during transmission and storage.

3.3.5 Usability

- The user interface should be intuitive and easy to navigate.
- Mobile responsiveness is essential to ensure usability on various devices, including smartphones and tablets.

3.3.6 Maintainability

- The codebase should be well-documented to facilitate future maintenance and updates.
- The system should support easy updates and feature additions without major downtime.

Chapter 4: Diagrams

4.1 ER Diagram

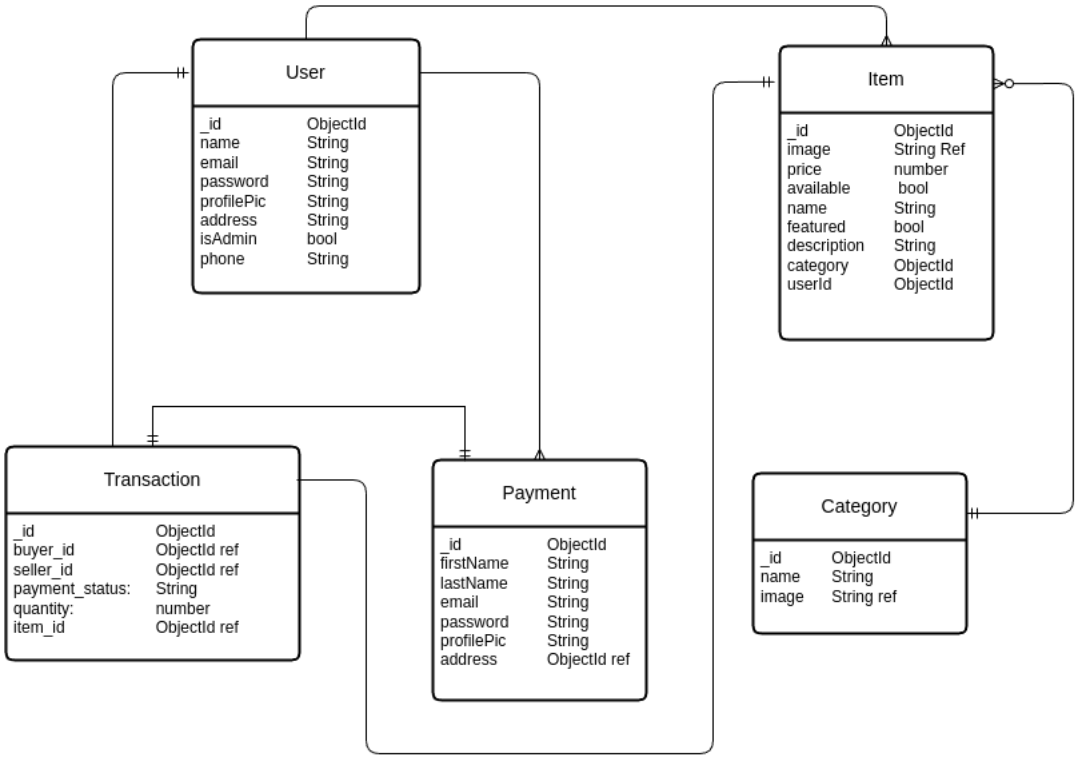

Figure 4.1: Entity-Relationship Diagram of ePurano

The provided diagram is an Entity-Relationship (ER) diagram that depicts the relationships between various entities in the ePurano application. Here's a detailed explanation of each entity and their relationships:

1. **User**
    - The User entity represents individuals who use the ePurano platform to buy or sell secondhand items. Each user has a unique identifier userId, and other attributes such as name, email, password and profileImage
    - A User can create multiple Items, meaning they can list several products for sale on the platform.
    - A User can also be involved in multiple Transactions, either as a buyer or a seller, reflecting their activity within the marketplace.
2. **Item**
    - The Item entity represents the secondhand goods listed for sale on ePurano. Each item has a unique identifier itemId and attributes such as title, description, price, and image.

- An Item is associated with one User, who is the seller listing the product. This is referenced by the userId foreign key.
- An Item belongs to one Category, linking it to a broader classification within the marketplace through the categoryId foreign key.
- An Item can be involved in multiple Transactions, indicating that it can be purchased multiple times by different users or repeatedly by the same user.

3. **Category**

   - The Category entity represents the classification of Items within the ePurano marketplace. Each category has a unique identifier categoryId and a name attribute.
   - A Category can have multiple Items, indicating that several products can belong to the same classification.

4. **Transaction**

   - The Transaction entity represents the exchange of an Item between a buyer and a seller. Each transaction has a unique identifier transactionId.
   - A Transaction involves one Item, linked through the itemId as a foreign key.
   - A Transaction involves one Payment, indicating the financial transaction associated with the exchange. This is represented by the paymentId foreign key.

5. **Payment**

   - The Payment entity represents the financial transaction for a purchase on ePurano. Each payment has a unique identifier paymentId and attributes such as amount, paymentDate, and user details.
   - A Payment is associated with one Transaction, indicating that each financial transaction corresponds to a single exchange of goods.

4.2 Dataflow Diagram

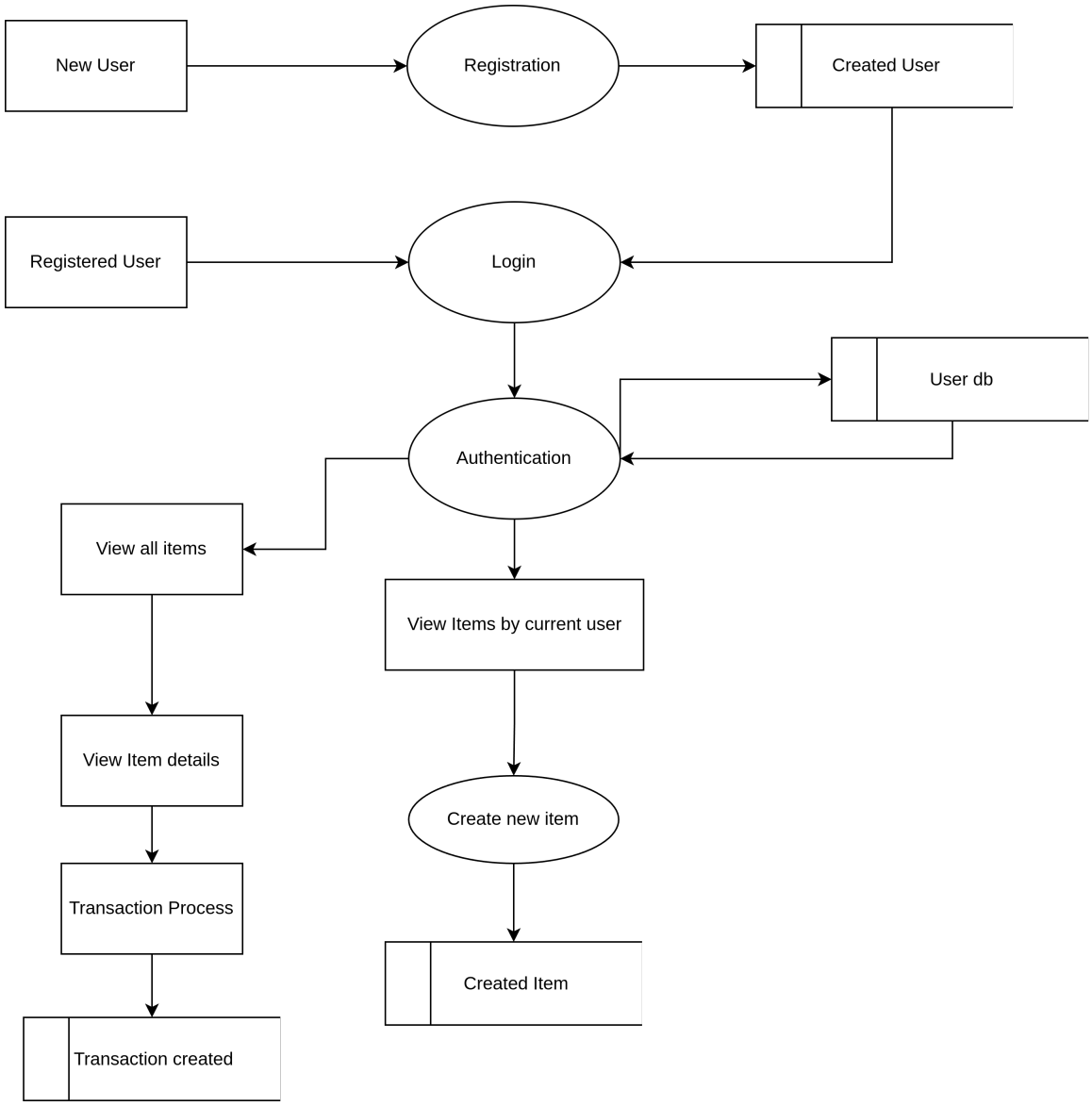

Figure 4.2: Dataflow Diagram of ePurano

The Data Flow diagram shows a high-level overview of how a data flows inside the system where a new user register or an already registered user log in. The new user goes through the registration process and verification process and after that the user record is saved to the database. The existing user log in by going through authentication process. After logging in, they can view different items and can create their own. The Payment is done by cash on delivery or using the Khalti online payment gateway.

4.3 State Diagram

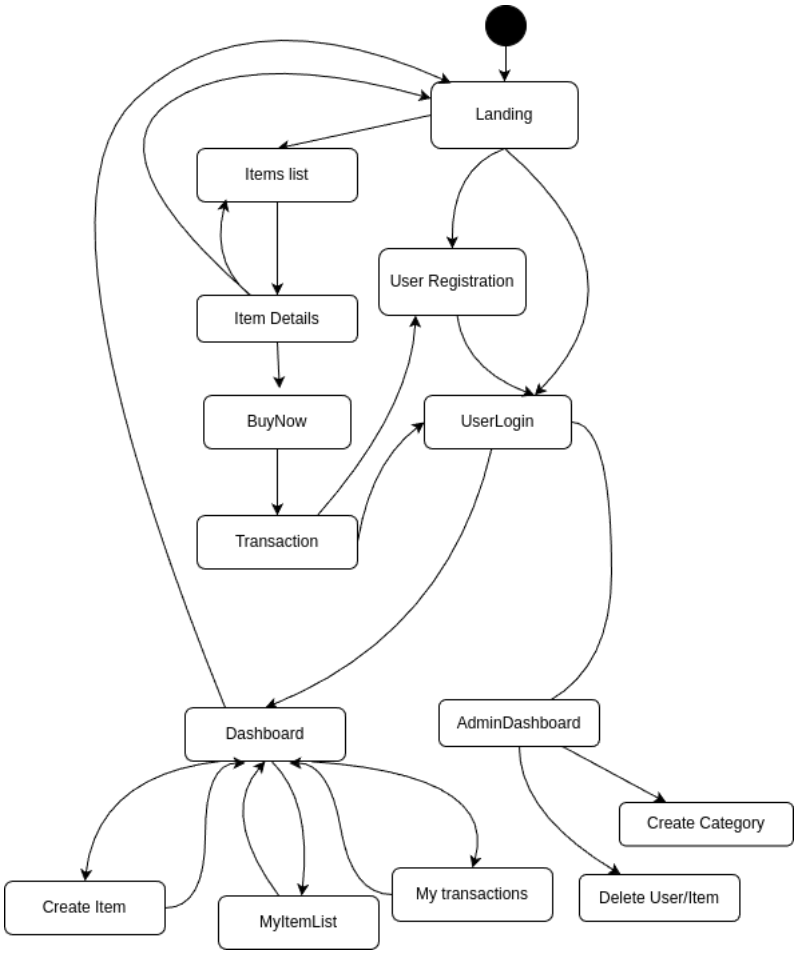

Figure 4.3: State Diagram of ePurano

The state diagram for the ePurano application illustrates the various states the system can be in and the transitions between these states based on user actions.

4.4 Use Case Diagram

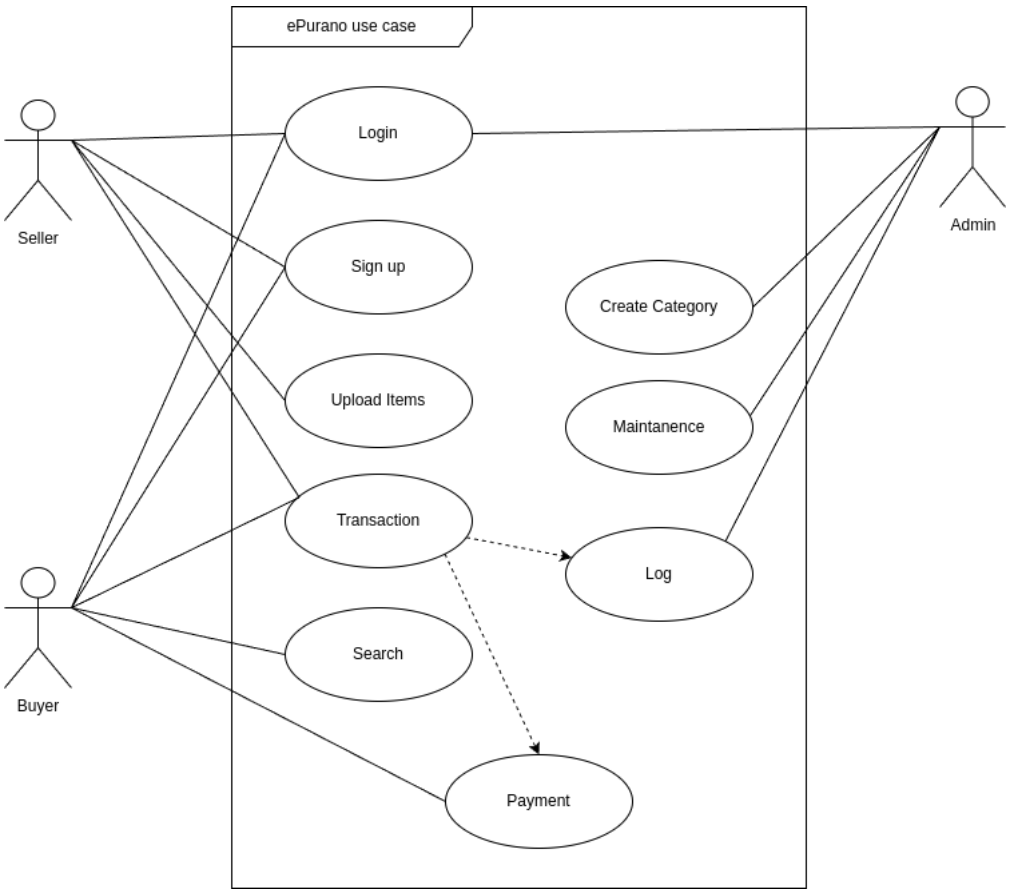

Figure 4.4: Use Case Diagram of ePurano

The use case diagram for the ePurano application illustrates the different interactions between users (both regular users and admins) and the system. Here's a detailed description of each use case and the associated actors:

1. **Login**

   Actor User, Admin

   Description Both users and admins can log in to the system to access their respective dashboards and functionalities.

2. **Register**

   Actor User

   Description New users can register an account on the platform to start buying and selling secondhand items.

3. **Create Item**

   Actor User

Description Registered users can upload details and images of the items they wish to sell.

4. **Transaction**

   Actor User

   Description Users can complete transactions for purchasing items listed on the platform.

5. **Create Category**

   Actor Admin

   Description Admins can create new categories for items, helping to organize listings and improve searchability.

Abbreviations

ePurano The name of the application platform

P2P Peer to peer

SRS Software Requirements Specification

SDLC Software Development Life Cycle

ER Entity Relation

API Application Programming Interface